Guía de ejercicios # 2

Organización de Computadoras 2016

UNQ

Arquitectura Q1

Características

- Tiene 8 registros de uso general de 16 bits: R0..R7

Instrucciones de 2 operandos

El siguiente es el formato de las instrucciones de **Q1**, las cuales tienen dos operandos (origen y destino). Los tamaños de los campos están expresados en bits.

Cod.Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Destino (16b)	Origen (16b)
----------------	----------------------	---------------------	------------------	-----------------

Con este formato, los campos **Destino** y **Origen** contienen valores constantes (si el modo respectivo es *inmediato*) o no existen (si el modo respectivo es *registro*).

A continuación la tabla con las operaciones disponibles:

Operación	Cod Op	Efecto
MUL	0000	Dest ← Dest * Origen
MOV	0001	Dest ← Origen
ADD	0010	Dest ← Dest + Origen
SUB	0011	Dest ← Dest - Origen
DIV	0111	Dest ← Dest % Origen

Nota: El carácter % denota el cociente de la división entera.

El resultado de la operación **MUL** ocupa 32 bits, almacenándose los 16 bits menos significativos en el operando destino y los 16 bits más significativos en el registro **R7**.

Modos de direccionamiento

Los modos de direccionamiento se codifican según la siguiente tabla:

Modo	Codificación
Inmediato	000000
Registro	100rrr

donde **rrr** es una codificación (en 3 bits) del número de registro.

Ejemplos de algunas instrucciones válidas:

- MOV R0,R1
- ADD R1,0x0000
- MUL R1,0x000A

Ejercicios

1. Escriba un programa que inicialice el registro **R0** con la cadena que representa el valor 3 en **BSS(16)**
2. Escriba un programa que inicialice el registro **R0** con la cadena que representa el valor 15 en **BSS(16)**
3. Escriba un programa que inicialice el registro **R0** con la cadena que representa el valor 16 en **BSS(16)**
4. Escriba un programa que inicialice el registro **R0** con la cadena que representa el valor 255 en **BSS(16)**
5. Escriba un programa que inicialice el registro **R0** con la cadena que representa el valor 256 en **BSS(16)**
6. Escriba un programa que inicialice el registro **R0** con la cadena que representa el valor 511 en **BSS(16)**
7. Escriba un programa que inicialice el registro **R0** con la cadena que representa el valor 512 en **BSS(16)**
8. Escribir un programa que duplique el valor contenido en el registro **R0**
9. Escribir un programa que le sume el valor 3 al contenido del registro **R1**
10. Escribir un programa que multiplique por 12 el contenido del registro **R0**
11. Considere el sistema **BSS(16)**
 - (a) Represente los números 115 y 87
 - (b) Realice la suma de las cadenas obtenidas en el punto anterior
 - (c) Interprete la cadena resultante de la suma
12. Escribir un programa que calcule el valor de la expresión $115 + 87$. **Nota:** No debe resolver la cuenta, sino hacer un programa que lo haga
13. Escribir un programa que calcule el valor de la expresión $60000 + 8000$. **Nota:** No debe resolver la cuenta, sino hacer un programa que lo haga
14. Escribir un programa que sume los valores de los registros **R1** y **R0**, y ponga el resultado en **R2** (sin modificar **R1** y **R0**)
15. Escribir un programa que a **R5** le reste 2 veces el valor que tiene **R6**

16. Escribir un programa que calcule el promedio entre los registros R2 y R3
17. Escribir un programa que a R4 le sume los valores de R1 , R2 y R3 ; y le reste los valores de R5 , R6 y R7

```
SUB R0, R1
ADD R2, R0
DIV R2, 0x0007
MUL R5, 0x000E
```

18. Escribir un programa que calcule el promedio entre los registros R2 , R3 y R4
19. Suponiendo que en el registro R4 tiene almacenado el costo en pesos de un producto, y el registro R5 contiene el porcentaje descuento a aplicar, calcule el valor final de venta a ser almacenado en R6 .
20. La siguiente afirmación, ¿es verdadera o falsa? *El código fuente de un programa debe estar en memoria para ser ejecutado.* Justifique

30. Desensamblar el siguiente programa:

```
0001100000100001
0000100001100000
0010100001000000
0000000000000101
```

21. Ensamblar el siguiente programa

```
ADD R0,0xFAFF
```

22. Ejecute el programa anterior, suponiendo que antes de comenzar el registro R0 contiene el valor 0x0001.

23. Ensamblar el siguiente programa

```
SUB R6,0x5678
```

24. Ejecute el programa anterior, suponiendo que antes de comenzar el registro R6 contiene el valor 0xFFFF.

25. Ensamblar el siguiente programa

```
ADD R0,R0
DIV R1,0x0003
MUL R0,R1
```

26. Ejecute el programa anterior, suponiendo que antes de comenzar el registro R0 contiene el valor 0x0001 y el registro R1 contiene el valor 0x000F. ¿Que registros fueron alterados? ¿Que valores tienen ahora?

27. Ensamblar el siguiente programa

```
ADD R7,0x000F
DIV R6,R7
SUB R6,R1
```

28. Ejecute el programa anterior, suponiendo que antes de comenzar el registro R7 contiene el valor 0x0001, el registro R6 contiene el valor 0x0AAA y el registro R1 contiene el valor 0x0A3B. ¿Que registros fueron alterados? ¿Que valores tienen ahora?

29. Ensamblar el siguiente programa: